



15  
**RECEIVED**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

AUG 21 1992

In re application of : Before the Examiner  
R. A. Fabbio et al : A. Jankus  
**GROUP 2300**

Serial No.: 07/352,530 : Group Art Unit: 2301  
Filed: 5/15/89 : Intellectual Property  
Title: OBJECT DATABASE-DRIVEN: Law Department  
INTERACTIVE SHELL FOR A : International Business  
DATA PROCESSING SYSTEM : Machines Corporation  
: 11400 Burnet Road  
: Austin, TX 78758

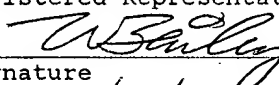
August 6, 1992 :

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to Attention: Commissioner of Patents and Trademarks, Washington, D. C. 20231 on August 6, 1992.

Wayne P. Bailey

Registered Representative, Reg. No. 34,289

  
Signature

8/6/92  
Date

TRANSMITTAL OF APPELLANTS' BRIEF UNDER 37 CFR 1.192(a)

Honorable Commissioner of Patent and Trademarks

Washington, D.C. 20231

Sir:

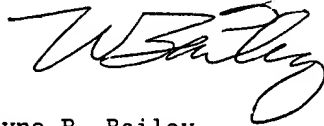
Attached is Appellants' Brief, in triplicate, from a decision of the Examiner dated February 12, 1992, finally rejecting Claims 1-27.

Please charge my Deposit Account No. 09-0451 in the amount of \$260.00.

The Commissioner is hereby authorized to charge any additional fee which may be required or credit any overpayment to

Deposit Account No. 09-0451. A duplicate copy of this document  
is included.

Respectfully submitted,

A handwritten signature in cursive script, appearing to read "W Bailey".

Wayne P. Bailey  
Attorney for Applicants  
Registration No. 34,289  
(512) 823-1012

WPB/bar



RECEIVED PATENT 07/352,530

AUG 21 1992 #15

GROUP 2300 103

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of : Before the Examiner:  
R. A. Fabbio et al : A. Jankus  
Serial No.: 07/352,530 : Group Art Unit: 2301  
Filed: 5/15/89 : Intellectual Property  
Title: OBJECT DATABASE-DRIVEN : Law Department  
INTERACTIVE SHELL FOR : International Business  
A DATA PROCESSING : Machines Corporation  
SYSTEM : 11400 Burnet Road  
August 6, 1992 : Austin, Texas 78758

94-1648

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231 on August 6, 1992.

Wayne P. Bailey  
Attorney for Applicants, Registration No. 34,289

Signature *W Bailey*

Date *8/6/92*

APPELLANT'S BRIEF UNDER 37 CFR 1.192

Honorable Commissioner of Patents and Trademarks

Washington, D.C. 20231

Sir:

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES  
U. S. PATENT AND TRADEMARK OFFICE

APPEAL FROM A DECISION OF THE  
EXAMINER FINALLY  
REJECTING CLAIMS 1-27 OF  
U. S. PATENT APPLICATION  
SERIAL NUMBER 07/352,530

SC13238 08/19/92 07352530

09-0451 130 120 260.00CH

AT9-89-039

1

BRIEF FOR APPLICANTS - APPELLANTS

(1)

Status of Claims

Claims 1-27 stand rejected under 35 U.S.C. § 103. Claim 27 stands rejected under 35 U.S.C. § 101. The application was filed with twenty-seven (27) claims and none have been added during prosecution. Claims 1-27 are set forth in the Appendix (section (7)) to this brief which follows immediately after the signature of Applicants' attorney.

(2)

Status of Amendments

The final rejection was dated February 12, 1992. A C.F.R. 1.116 amendment after final was filed April 15, 1992. The Examiner considered this amendment after final, but determined that it did not place the application in condition for allowance and did not enter it.

(3)

Summary of the Invention

The system and method of this invention are an interface tool (10 of Figure 1B) for displaying menus and dialogues to a user of a data processing system. The actual visual presentation to the user is performed by any screen library using the data from this interface tool. The interface tool of this invention is a generic engine that has no knowledge of the individual instances of system resources that it presents to a user. The navigation through menus, and presentation and collection of data are all directed toward the goal of calling an appropriate function in the application, or system, layer (20 of Figure 1B) of a computer operating system. This work can be done without special knowledge because the interface

tool is driven by self-defining, independent data objects which are sufficient to give direction to the interface layer for the accomplishment of the above tasks.

The menus, dialogues, and each instance of a system resource are represented as objects within an object database, and are referred to as interface objects (30 of Figure 1B), as described at Specification page 10, lines 19-21. Data within these interface objects (and depicted in Figures 2, 3 and 6) reflect the topology of the system resources, as shown by Figure 7. The interface tool traverses these interface objects based upon the data within the interface objects themselves and the user selections, as described at Specification page 10, line 23 through page 11, line 2. Hence, the interface tool is data driven, and not predefined within programming code (Specification page 42, line 1 through page 43, line 12).

The interface data objects define the workings of the interface, are installed with the software (applications) that they control, and are independent of other layers within the data processing system, as described at Specification page 13, lines 15-26, page 14, lines 14-27 and page 22, line 32 through page 23, line 24. The interface tool (10) gets information that it needs, which is contained in the system resource data objects (30), via calls to functions in the application layer (20) rather than by direct reference to this body of data.

The interface tool can independently execute any command, controlling all input and output (Specification page 33, line 22 through page 36, line 23). It therefore has access to, and control of, all system resources by execution of the appropriate command shell functions, as described at Specification page 25, line 27 through page 26, line 14.

This object-oriented interface model allows for highly flexible, easily extendible interfaces. Pieces of the interface can be added or subtracted with minimum impact to the layer as a whole, as described at Specification, page 12, lines 3-12. With the addition of each new piece of software, interface objects can be added to the data repository, allowing newly installed parts of the system to be managed in exactly the same way as parts that were the first to be designed, as described at Specification page 26, lines 15-33. Hence, there is no need for third party vendors to negotiate with the original manufacturer to get the support they need for their application. The problem of maintenance is diminished because the user who is extending or tailoring her/his user interface does not edit large amounts of data (tables, programs, etc.), but rather creates new interface objects and adds them to the ones already in the repository. Without the need of re-compilation (since the system is data driven, and not predefined in the programming code), the new objects appear in the desired environment the next time that the environment is entered because they will share a common id key with the other objects in that environment and therefore be found by the interface tool. Specific examples of this capability can be found at Specification page 12, line 18 through page 13, line 14 (with reference to Figures 10A, 10B and 10C) and page 43, line 13 through page 44, line 27 (with reference to Figures 1B, 2, 7 and 8).

(4)

Issue

The first issue is whether Claims 1-27 were properly rejected by the Examiner under 35 U.S.C. 103.

The second issue is whether Claim 27 was properly rejected by the Examiner as being unpatentable subject matter under 35 U.S.C. 101.

(5)

Grouping of Claims

The rejected claims do not stand or fall together, and Applicants consider the following groups of claims to be separately patentable:

Group I:	Claims 1, 3, 13-17, 24 and 26
Group II:	Claims 2, 6-8
Group III:	Claim 4
Group IV:	Claim 5
Group V:	Claim 9
Group VI:	Claims 10 and 11
Group VII:	Claim 12
Group VIII:	Claim 18
Group IX:	Claim 19
Group X:	Claim 20
Group XI:	Claim 21
Group XII:	Claims 22 and 23
Group XIII:	Claim 25
Group XIV:	Claim 27

(6)

Argument

Claims 1-27 stand rejected under 35 U.S.C. 103 as being unpatentable over Beck et al. Claim 27 stands rejected under 35 U.S.C. 101 as being non-statutory subject matter.

Regarding the 35 U.S.C. 103 rejection, the Examiner has stated that Beck teaches "interface object", "dynamically associating", and "based upon the data". The Examiner further

stated that the graphical representations of objects taught by Beck could be construed as interface objects, and that "dynamically associating" could be reasonably be interpreted as a mere frame response to a user selected object.

As specifically described below with respect to the individual claims, Applicants maintain that the Examiner has erred in rejecting these claims as a matter of law. First, the Examiner has failed to make a prima facie showing of obviousness. Secondly, the Examiner has failed to show a suggestion or motivation, in the single reference which has been cited by the Examiner, to substantiate the Examiner's assertion of obviousness.

A proper analysis of claims under 35 U.S.C. 103 begins with the question of whether the prior art made obvious the invention as a whole. Hartness International Inc. v. Simplimatic Engineering Co., 819 F.2d 1100, 1108, 2 USPQ2d 1826, 1832 (Fed. Cir. 1987). To aid in determining obviousness, inquiry must be made as to (1) the scope and content of the prior art, (2) the differences between the prior art and the claims at issue; (3) the level of ordinary skill in the art; and (4) so-called "secondary considerations", such as commercial success, long-felt need, or unexpected results. Graham v. John Deere Co., 383 U.S. 1, 17, 148 USPQ 459, 467 (1966).

Applicants now show that the claims of Group I (Claims 1, 3, 13-17, 24 and 26) would not be obvious in view of Beck. Beck fails to teach or suggest 'dynamically associating ... based upon the data within each of said ... interface objects'. Such claimed element is a key aspect of the claimed invention. As Applicants' claimed objects allow for dynamic association based upon data contained within such object, the objects are active in nature. Beck's graphical



representations are passive, and merely depict a transmitted and receiving object. This key component of the claimed invention is not disclosed by Beck, nor has the Examiner so alleged. The invention would not have been obvious in light of the prior art because the single cited reference does not disclose or suggest this feature of Claim 1. Nor does the reference suggest modification in such a way to achieve Applicants' claimed invention. One of the differences between Claim 1 and the considered art is this critical missing feature. Thus, a person of ordinary skill in the art would have no teaching or suggestion in the references of Applicants' claimed invention. Symbol Technologies Inc. v. Opticon Inc., 19 USPQ 1241, 1247 (Fed. Cir. 1991). The mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification. In re Mills, 16 USPQ2d 1430, 1432 (Fed. Cir. 1990).

Applicants further show that their claimed invention in Group I is not obvious in view of Beck. The Beck system, when triggered by a message transmission, displays representations of objects comprising a box with labels identifying the represented object (Beck Col. 3, lines 2-6). There is no suggestion or teaching in Beck of using data within the object itself to aid, assist, or be used in conjunction with the dynamic association of objects with the screen representation. If Beck has any dynamic association at all, it is the use of messages to invoke screen representations of objects. These Beck messages are not contained within the objects to be displayed by a screen representation. Rather, the messages are used to convey information between, and thus outside of, the objects.

To reiterate, Applicants are claiming the use of data within the object to dynamically associate objects with frame presentations. This provides greater extendibility and flexibility in adding objects and corresponding screen representations for such objects, as the interface objects can be added to or deleted from the system independently of one another. Recompile of the menu tool is not necessary when adding or deleting objects, as the claimed invention is data driven, and not predefined within programming code. This flexibility simply does not exist in the teachings of Beck.

Claims 2-25 are similarly allowable, as they contain all the limitations of Claim 1 which has been shown to be allowable in view of Beck. However, notwithstanding the above, Applicants will further show how these claims are allowable in view of Beck.

Regarding Group II (Claims 2, 6-8), the Examiner rejected Claim 2 in stating that Beck can reasonably be interpreted to teach the use of "attributes of system resources". The Examiner has failed to show where Beck discusses or teaches using objects to represent attributes of system resources as claimed by Applicants, and has thus failed to make a prima facie showing of obviousness. Beck's objects do not represent the particular attributes of system resources. This aspect of Applicants' claimed invention allows for a user to easily manage system resources in a data processing system, by providing the ability to easily add, delete, or modify menu interactions with a user that correspond to such system resources without undue burden or operating system regenerations. The teachings of Beck do not even address a similar problem, much less teach the solution being claimed by Applicants. Beck addresses the problem of debugging software, not of managing system resources. As the problems and solutions

of Beck do not relate or pertain to Applicants' claimed invention, a person of ordinary skill in the art would not be motivated to adapt the teachings of Beck to that being claimed by Applicants. Diversitech Corp. v. Century Steps, Inc., 850 F.2d 675, 7 USPQ 1315 (Fed. Cir. 1988). Thus, Claim 2 was improperly rejected, as a prima facie case for obviousness has not be made. In re Gordon, 221 USPQ 1125 (Fed. Cir. 1984).

For Group III (Claim 4), the Examiner maintains the rejection of Claim 4, which was previously rejected for reasons given in Claim 1. Applicants maintain that Beck does not teach any type of dynamic association using hierarchical data stored within an object, as is being claimed by Applicant. Beck uses messages to trigger objects, and the methods have no hierarchical data, nor are they a part of the objects themselves. The hierarchical aspect of Applicants' claimed invention provides for the grouping and maintaining of objects, which allows ease in management and control of such objects and the corresponding entities they represent. Claim 4 has been improperly rejected, as all the claimed elements have not been shown in the art of record, nor has there been a showing of suggestion or motivation disclosed within the art of record which would enable a person of ordinary skill in the art to achieve Applicants' claimed invention..

Group IV (Claim 5) has the limitation of 'managing of a screen presentation of the objects and a user interaction with said objects based upon data within ... interface objects'. This element is not taught or suggested by Beck. Beck's graphical representations are mere passive, output elements (Beck, Col. 4, lines 26-30). No use is made of data within the objects for assistance in the management of the screen presentation, as is being claimed by Applicants. Applicants' claimed limitation of managing the screen presentation based

upon data within an object is a key feature of Applicants' claimed invention. This feature provides for ease in system extensions or modifications to be made to the user interface by merely adding additional objects. The nature of the problem which persisted in the art, and the inventor's solution, are factors to be considered in determining whether the invention would have been obvious to a person of ordinary skill in the art. Northern Telecom Inc. v. Datapoint, 15 USPQ2d 1321, 1324 (Fed. Cir. 1990). Beck fails to even address the problem of flexibility in extending the user interface, much less teach a solution to the problem which Applicants have solved. Therefore, Claim 5 would not have been obvious in view of Beck, as the problems and solutions facing Beck differ from those of the Applicants. Further, there is no teaching or suggestion to modify the reference to achieve Applicants' claimed invention. Symbol Technologies Inc. v. Opticon Inc., supra.

Regarding Group V (Claim 9), Applicants recite 'means for utilizing a current value of said... attribute of said ... system resource for a validation of a user response'. Beck does not teach system resources having attributes, or current values of attributes being used for validation of user responses. These claimed features solve the problem of providing ease of validating user input during system operation. Again, the problem facing Applicants, and Applicants' resultant claimed solution were not discussed or otherwise addressed by the teachings of Beck. Therefore, the limitations claimed by Applicants in Claim 9 are patentable in view of Beck, as a person of ordinary skill in the art would not have been motivated to modify the teachings of Beck to achieve Applicants' claimed invention. Claim 9 was thus improperly rejected, and should be allowed.

Applicants recite in Group VI (Claims 10 and 11) a way of constructing a command based upon an input value and an option contained within an interface object. Beck does not disclose this claimed limitation. It has no teachings of building commands based upon an option contained within an interface object. The command line invocation of Beck in no way teaches or suggests the limitations being claimed by Applicants. Beck merely displays and executes a command, but does not construct a command. Claim 10 further includes the limitation that the command construction occurs dynamically, as a result of user interaction and an interface object option. The command is not merely static in nature, as is the Beck command(s). Claim 10 is allowable in view of Beck, and has been improperly rejected, as there are missing claimed features in the cited reference. Further, there is no teaching or suggestion within Beck to enable a person of ordinary skill in the art to achieve Applicants' claimed invention. Symbol Technologies Inc. v. Opticon Inc., supra

Regarding Group VII (Claim 12), Beck does not teach logging commands for later execution. Beck teaches displaying a list of messages in progress. This is not what is being claimed by Applicants, who are claiming 'logging said command for later execution'. This logging is a deferral method, not a status indicating method of Beck. Logging provides the added benefit of being able to subsequently configure a system identical with a previously configured system, without having to repeat the user interaction with the menus. Again, Beck fails to even address the problem of providing such a fast-path means of executing a previously built command, nor does Beck teach or suggest the solution claimed by Applicants. Thus, Claim 12 has been improperly rejected and is allowable in view of Beck.

Group VIII (Claim 18) includes the limitation of 'altering an object database from within the interface during a session of execution ... and ... reflecting said altered interface during said same session' and is nowhere taught or suggested by Beck. This is a unique capability of Applicants' claimed invention, where the underlying database can be modified in real time, without the need for system regeneration(see Specification, page 8, line 21 to page 9, line 6). Thus, Claim 18 was improperly rejected as there is no teaching or suggestion within Beck of this unique claimed feature.

Group IX (Claim 19) includes the limitation of altering an interface object database by creating a new interface object. This is not the same or similar to the Beck teachings of updating a visual display, which is not an object database. There is no connection between Beck's status information being displayed, and the ability to alter an underlying object database. Therefore, Claim 19 was improperly rejected and should be allowed as there is no teaching or suggestion within Beck of this unique claimed feature.

Group X (Claim 20) includes the limitation of directly entering a hierarchy of objects. There is no discussion, teaching, or suggestion of directly entering a hierarchical relationship of interface objects by Beck. Beck merely teaches a fixed hierarchy of classes, and the ability to suppress the display of intermediate messages. Therefore, Claim 20 was improperly rejected by the Examiner in view of Beck, and should be allowed, as a prima facie case for obviousness has not been made by the Examiner. Further, there is no teaching or suggestion within Beck to enable a person of ordinary skill in the art to achieve Applicants' claimed invention.

Group XI (Claim 21) was rejected by the Examiner stating that Beck teaches means for displaying presentations by a plurality of graphical libraries. However, there is no teaching or suggestion for 'displaying said logical frame presentations by a plurality of graphical libraries', as is claimed by Applicants. This support for plural graphical libraries is another key feature of Applicants claimed invention, and allows for future applications to use graphical libraries which are supplied by the application, and bypass any existing graphical libraries predefined by the system. This additional degree of flexibility in the underlying system design is in no way taught or suggested by Beck. There is no teaching or suggestion of a graphical library, nor is there a teaching or suggestion supporting a plurality of graphical libraries. The Examiner states that Beck's teaching of a message-set browser is a reasonable interpretation of graphical libraries. The Examiner is apparently equating the use of multiple windows with multiple graphical libraries. As is known to those of ordinary skill in the art who would reasonably interpret the scope of the claimed element 'graphical libraries', this does not refer to the display of multiple windows, i.e. "libraries" does not mean "windows". Thus, Claim 21 was improperly rejected by the Examiner, and is allowable in view of Beck. There is no teaching or suggestion within Beck which would motivate a person of ordinary skill in the art to implement Applicants' claimed invention.

In Group XII (Claims 22 and 23), Applicants are claiming 'means, within said interface objects, for representing...'. Beck's graphical representations have no means to do anything. They are mere output representations, and contained no information within themselves, for representing items in a logical frame in a plurality of ways depending upon a graphical or

linguistic context, as is claimed by Applicants. Claims 22 and 23 have been improperly rejected and should be allowed, as a prima facie case for obviousness has not been made by the Examiner. Further, there is no teaching or suggestion within Beck to motivate a person of ordinary skill in the art to achieve Applicants' claimed invention.

Group XIII (Claim 25) includes the limitation of an access control policy. No access control policy is taught or suggested by Beck. Rather, Beck merely displays a list of commands for a user to invoke, and has no means for providing any access control policies on commands available for selection. This access control policy capability greatly enhances the usability of the claimed system in a secured environment. Furthermore, Beck fails to address the problem of control policies for interface objects. Claim 25 was improperly rejected and should be allowed, as a prima facie case for obviousness has not been made by the Examiner. Further, there is no teaching or suggestion within Beck to motivate a person of ordinary skill in the art to achieve Applicants' claimed invention.

Group XIV (Claim 27) is allowable for reasons given above regarding Group I.

#### CONCLUSION FOR 35 U.S.C. 103 REJECTION

Regarding the rejection under 35 U.S.C. 103, it is emphasized that the Beck graphical representations are mere passive output indicators and have no bearing or relationship to the interface objects being claimed by Applicants. Applicants' interface objects have information contained within the objects themselves, and this information and objects are used to drive (i.e. is an active input to) a target system resource. There is no teaching within Beck that provides this



missing element, nor is there a suggestion or other motivation within Beck to provide this missing element. As the Beck graphical representations and Applicants' interface objects have no bearing or relationship to one another, it would further not be obvious to modify the teachings of Beck to achieve Applicants' claimed invention in Claims 1-27. Additionally, the problems facing Applicants (ease in system configuration/reconfiguration, and corresponding user interface menus, when modifying system resources) are totally distinct from those which Beck faced (debugging software), which further evidences non-obviousness.

For all the above reasons, Applicants maintain that these Claims 1-27 have been erroneously rejected by the Examiner under 35 U.S.C. 103.

#### 35 U.S.C. 101 REJECTION

The Examiner rejected Group XIV (Claim 27) under 35 U.S.C. § 101 as being "directed to non-statutory subject matter".

Section 101 of the Patent Act describes the range of subject matter that is eligible for patent protection. It states that anyone who "invents or discovers any new and useful process, machine, manufacture, or composition of matter" may obtain a patent for this innovation<sup>1</sup>. The legislative history of 35 U.S.C. § 101 demonstrates that Congress intentionally worded the section broadly so it would encompass wide areas of unforeseen invention<sup>2</sup>.

---

<sup>1</sup>35 U.S.C. § 101 (1976).

<sup>2</sup>S. Rep. No. 1979, 82d Cong., 2d Sess. 5 (1952), reprinted in 1952 U.S. Code Cong. and Admin. News 2394, 2399; H.R. Rep. No. 1923, 82d Cong., 2d Sess. 6 (1952).

An invention may be patented only if it falls within one of the four statutory classes of subject matter of 35 U.S.C. § 101. Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470 (1974). In construing 35 U.S.C. § 101, the Supreme Court in Diamond v. Diehr, 450 U.S. 175 (1981) and Diamond v. Chakrabarty, 447 U.S. 303 (1980), has applied a broad interpretation to statutory subject matter so as "to include anything under the sun that is made by man". Any process, machine, manufacture, or composition of matter constitutes statutory subject matter *unless* it falls within a judicially determined exception to section 101. In re Pardo, 684 F.2d at 916, 214 USPQ at 677 (CCPA 1982). The major exception, if not only exception, in the area of computer processes is the mathematical algorithm. Other areas of judicially defined exceptions to patentability under 35 U.S.C. § 101 include laws of nature, physical phenomenon, and abstract ideas. Diamond v. Diehr, 450 U.S. 175 (1981). The issue is thus whether a computer program, residing on a computer readable medium, has been judicially recognized as being non-statutory subject matter. If not, Applicants are entitled to a patent as a matter of right<sup>3</sup>.

In rejecting Claim 27, the Examiner stated that the claimed subject matter does not fall within any of the statutory classes listed in 35 U.S.C. 101, but rather recites a computer program.

---

<sup>3</sup>The patent statutes give to inventors the right to a patent upon compliance with their provisions, and neither the rules promulgated by the Patent Office nor the interpretation placed upon them can detract from these rights. In re Stempel, 113 USPQ 77, 81 (CCPA 1957). Under 35 U.S.C. 102 an applicant is "entitled to a patent unless" it is shown that one or another of the prohibitory provisions therein, or elsewhere in the statute, applies. Id.

#### ABSTRACT IDEA

It is a fundamental precept of patent law that patents are not issued for abstract ideas, but rather for new *means* to achieve useful results<sup>4</sup>. A mere "idea" is not patentable, but only because it is not "reduced to practice" as a specific means for achieving a useful result. Applicants are not claiming an abstract concept, but rather the specific means for presenting items for selection by a user in a data processing system. The claims are drafted in "means for" language, as allowed by 35 U.S.C. § 112, sixth paragraph, and encompass the corresponding structure, material, or acts in support thereof<sup>5</sup>. Such claimed limitations include, for example, (i) means for representing a plurality of interface objects in a object database and (ii) means for dynamically associating different ones of the interface objects with a plurality of logical frame presentations. Applicants are reciting a combination claim, i.e. a combination of elements or means which together result in the desired result. Each of the recited means reside on a computer compatible medium, which specifically recites a structural limitation. Thus, the claimed means would not preempt any abstract idea, mathematical algorithm or other law of nature.

#### COMPUTER PROGRAMS

Next, Applicants will address whether computer programs are *per se* non-statutory, notwithstanding the fact that the claims are drafted in a form to overcome the abstract idea

---

<sup>4</sup>1 D. Chisum, Patents: A Treatise on the Law of Patentability, Validity and Infringement, § 1.03[2] (Rel. 17 1986).

<sup>5</sup>35 U.S.C. § 112 (1982).

rejection. As stated in In re Gelnovatch and Arell, 201 USPQ 137, 141 (CCPA 1979)<sup>6</sup> "the mere labelling of an invention as a 'computer program' does not aid in decision making". The courts have *not* held that all computer programs are unpatentable<sup>7</sup>, but only those which claimed a mathematical algorithm and in which the claim would effectively preempt an idea<sup>8</sup>. A claim drawn to subject matter otherwise statutory does not become non-statutory simply because it uses a mathematical formula, computer program, or digital computer. Diamond v. Diehr, 450 U.S. 175 (1981). The Examiner's rejection based upon the claimed invention being a "computer program" is thus made contrary to judicial interpretation of 35 U.S.C. § 101. This is a further conclusive showing that these claims have been erroneously rejected.

It should also be noted that the U.S. Patent and Trademark Office (hereinafter USPTO) explicitly established a subclass for software programs. Class 364, subclass 300, for Programming Methods and Procedures (which was maintained by the USPTO as recently as 1990) was defined as "Subject matter ... which includes software systems(programs) used in programmable digital data processing systems or computers for operating on digital data".

---

<sup>6</sup>This case was decided after the Benson decision.

<sup>7</sup>As stated in Gottschalk v. Benson, 409 U.S. 63 (1972), "It is said that the decision precludes a patent for any program servicing a computer. We do not so hold." See also In re Bradley and Franklin, 600 F.2d 807, 202 USPQ 480 (CCPA 1979), *aff'd sub nom.* Diamond v. Bradley, 450 U.S. 381 (1981).

<sup>8</sup>see, e.g. Gottschalk v. Benson, 409 U.S. 63 (1972), Parker v. Flook, 437 U.S. 584 (1978), Diamond v. Diehr, 450 U.S. 175 (1981).

Applicants will now show that a computer program residing on a computer readable medium is both a "subcombination" of a machine and an article of manufacture, which are specifically listed as being statutory under 35 U.S.C. § 101.

#### MACHINE

Inventions pertaining to machines ... may be divided into four classes<sup>9</sup>. The first class is where the invention embraces the entire machine. The second class of inventions are those which embrace one or more elements of the machine, but not the entire machine<sup>10</sup>. The U.S. Supreme Court has explicitly acknowledged the availability of patent claims to machine subcombinations<sup>11</sup>.

Applicants' computer program claims are directed to a machine subcombination, i.e. the medium containing the recited "means for" limitations. The Examiner's difficulty in analyzing Applicant's claims may be in the misunderstanding of what a computer program is or is not. Software is unlike almost any other engineering invention. First, software can be represented as both symbolic and mechanical<sup>12</sup>. As such, software symbology can be used by computer programmers to describe the program, and software mechanical properties are used to directly operate the computer. Software is thus both an

---

<sup>9</sup>Union Sugar Ref. v. Matthiessen, 14 F.Cas. 686 (No. 14,399) (D. Mass. 1865), *reprinted in part in* Chisum, supra at 1-10.

<sup>10</sup>Id.

<sup>11</sup>Denial of a patent on the subcombination would deprive the inventor of the benefit of the exclusive right to use the subcombination in the ways specified by the patent laws. Special Equipment Co. v. Coe, 324 U.S. 370 (1945).

<sup>12</sup>Davidson, Protecting Computer Software: a Comprehensive Analysis, 1983 Ariz. St. L.J. 611, 616 (1983).

engineering product and the specifications for that product. As an engineering product, it is suitable for patent protection, while as a written specification, it may qualify for copyright protection<sup>13</sup>. Software is also distinguished from computer programs. The following definition of "computer software" was adopted, after extensive debate, by both the World Intellectual Property Organization (WIPO)<sup>14</sup> and the Association of Data Processing Service Organizations (ADAPSO)<sup>15</sup>:

(i) "Computer program" means a set of instructions capable, when incorporated in a machine-readable medium, of causing a machine having information-processing capabilities to indicate, perform, or achieve a particular function, task or result;

(ii) "Program description" means a complete procedural presentation in verbal, schematic or other form, in sufficient detail to determine a set of instructions constituting a corresponding computer program;

(iii) "Supporting material" means any material, other than a computer program or a program description, created for aiding the understanding or application or

---

<sup>13</sup> Id.

<sup>14</sup> WIPO, Model Provisions On The Protection of Computer Software, § 1, Geneva (1978).

<sup>15</sup> ADAPSO, A Proposal For The Improved Protection Of Software, § 101, Software Protection Comm., Washington, D.C. (copies distributed April 1982).

application of a computer program, for example program descriptions and user instructions;

(iv) "Computer software" means any or several of the items referred to in (i) to (iii).

Applicants are claiming a computer program, as defined in (i) above, which is incorporated in a machine readable medium. This set of instructions could reside on such things as a magnetic diskette, magnetic tape, optical disk, Read Only Memory, Random Access Memory, Direct Access Storage Device, etc., i.e. any embodiment for which a computer could read the computer's instructions. The medium is devised, made, and used as a component part of a machine utilizing optics, magnetics and/or electronics to perform functions. Medium indicia for directing a machine's operations have been explicitly found to be patentable, as shown by Ex parte Lang, 56 USPQ 423 (Patent Office Board of Appeals 1943)<sup>16</sup> and In re Jones, 153 USPQ 77 (CCPA 1967)<sup>17</sup>. Applicants are not claiming a medium having random indicia thereon, but rather a medium where the novel and nonobvious selection and *arrangement* of the indicia, when used by a machine, results in specific novel and nonobvious claimed operations. This unique arrangement of indicia on the medium contain sequences of instructions that direct computers to perform various tasks.

Computer programs are the heart of computers. A computer without any programming of any kind is simply a lifeless form

---

<sup>16</sup>Punched card for controlling machine operation.

<sup>17</sup>Disk with series of patterns for controlling machine operation.

incapable of any task<sup>18</sup>. Since computers require computer programs to perform any useful function, the computer program itself is a subcombination of the machine. The above discussion conclusively shows that the claims are statutory as being directed to a machine subcombination.

#### MANUFACTURE

The claims are further directed to an article of manufacture. It is commonly known, and also set forth below, that a computer program, residing on a computer readable medium, is a tangible good, or article of manufacture. As such, this article of manufacture is statutory, per 35 U.S.C. § 101. The U.S. Court of Appeals for the 3rd Circuit stated in Advent Systems Limited v. Unisys Corporation, 925 F.2d 670, 674-75 (3d Cir. 1991)

software refers to the medium that stores input and output data as well as computer programs... Programs are codes prepared by a programmer that instruct the computer to perform certain functions. When the program is transposed onto a medium compatible with the computer's needs, it becomes software... That a computer program may be copyrightable as intellectual property does not alter the fact that once in the form of a floppy disc or other medium, the program is tangible, moveable and available in the marketplace. The fact that some programs may be tailored

---

<sup>18</sup>Gemignani, Legal Protection for Computer Software: The View from 1979, 7 Rut. J. Computers, Technology and the Law 269, 271.



for specific purposes need not alter their status as "goods".

It is clear from the above discussion that since a computer program residing on a computer readable medium is a "good", it is similarly an article of manufacture which falls under the gamut of allowable subject matter under 35 U.S.C. § 101.

#### MATHEMATICAL ALGORITHM

As stated in the M.P.E.P.<sup>19</sup> Section 2106, "the CCPA's two-step procedure set forth in In re Freeman, 197 USPQ 464 (CCPA 1978) is an appropriate test for determining if a claim involving mathematics and/or computer programming is in compliance with 35 U.S.C. § 101". Applicants now show that the claimed subject matter does in fact pass this two-step test.

To assist in determining whether a mathematical algorithm falls within the judicially defined mathematical algorithm exception<sup>20</sup>, the courts have adopted a two-step analysis against the claim at issue. The first step of the Freeman test, 573 F.2d 1237, 197 USPQ 464, as modified by In re Walter, 618 F.2d 758, 205 USPQ 397 and In re Abele, 684 F.2d 902, 214 USPQ 682 (hereinafter the Freeman-Walter-Abele test), is to determine if a mathematical algorithm is recited in the claim, either directly or indirectly. If a mathematical

---

<sup>19</sup>Manual of Patent Examining Procedure (MPEP), U.S. Dept. of Commerce, Rev. 13, November 1989.

<sup>20</sup>The exception applies only to *mathematical* algorithms since any process is an "algorithm" in the sense that it is a step-by-step procedure to arrive at a given result. In re Walter, 618 F.2d 758, 764 n.4, 205 USPQ 397, 405 n.4 (CCPA 1980); In re Pardo, 684 F.2d at 915, 214 USPQ at 676 (CCPA 1982); In re Iwahashi, 12 USPQ2d 1908 (Fed. Cir. 1989).

algorithm is found, the second step of the two-part test is applied. This second step is dependant upon the type of claims at issue, whether a method, apparatus, or "means for" type of claim as described at 35 U.S.C. § 112, 6th paragraph. If a claim does not contain a mathematical algorithm in the Benson sense<sup>21</sup>, the second step of the Freeman-Walter-Abele test is not reached<sup>22</sup>, and the claims are statutory under 35 U.S.C. § 101.

The claim on appeal does not recite a mathematical algorithm, either directly or indirectly. A claim should be considered as reciting a mathematical algorithm only if it essentially recites, directly or indirectly, a method of computing one or more numbers from different set of numbers by performing a series of mathematical computations. Ex parte Logan, 20 USPQ2d 1465 (Board of Patent Appeals and Interferences 1991). As described above, and further shown in the attached Appendix, the recited subject matter is directed towards specific "means" for performing specific non-mathematical operations, the "means" residing on a computer compatible medium. The concern of the Supreme Court in establishing this mathematical algorithm exception is to preclude a patentee from effectively preempting the use of the mathematical algorithm. There is no concern of preemption in this case, as the claims are limited to a particular computer readable embodiment. Applicants have thus successfully shown that analysis under the first step of the Freeman-Walter-Abele

---

<sup>21</sup>A mathematical algorithm is a "procedure for solving a given type of mathematical problem.". Ex parte Logan, 20 USPQ2d 1465 (Board of Patent Appeals and Interferences 1991), citing Gottschalk v. Benson, 409 U.S. 63 (1972); Diamond v. Diehr, 450 U.S. 175 (1981).

<sup>22</sup>1106 Official Gazette 5, 8 (1989).

test shows that no mathematical algorithm is recited in the claim under appeal. The second step of the analysis is not applicable, as there is no recitation of mathematical algorithms or preemption thereof. The claimed invention is thus shown to be statutory in view of the judicially defined mathematical exception to 35 U.S.C. § 101.

#### POLICY

The basic purpose of the patent system is to encourage the production and disclosure of new knowledge by offering inventors an opportunity to recover the costs of inventing. The patent system achieves this purpose by granting inventors exclusive rights to preclude others from making, using, or selling their claimed invention for a limited period of time. Society benefits from patent grants by receiving access to the disclosure of the patented invention. Because of the disclosure requirements of the patent laws, patent protection of software inventions allows for an increase in technological knowledge in the software industry. Disclosure makes this information more freely available for study and use. This new knowledge spreads throughout the software industry with increased speed and triggers new software inventions. This software innovation helps society use scarce resources more efficiently, as wasteful duplication of effort is eliminated.

The technological advancement in the computer industry that patent protection of computer programs encourages thus improves the general standard of living in America. Several empirical studies have concluded that technological advancement is a major reason for growth of per capita income in the

Western world over the last few centuries<sup>23</sup>. In addition, if modern American society continues its trend of emphasizing the importance of information processing and data collection, the growth and health of America's economy increasingly will depend upon advances in computer software technology<sup>24</sup>.

Patent protection of computer programs insures technological advancement in this area. Valuable inventions should be given protection of value in the real world of business and the courts<sup>25</sup>.

No basis exists for a moratorium on protection of inventions embodying or using computer programs. In re de Castelet, 195 USPQ 439 (CCPA 1977). Such broad prohibition could subject meritorious statutory inventions to unabatable piracy, and could forestall invention disclosure, the hallmark of the patent system. Id. Any justification for barring the issuance of computer software patents because of high administrative costs<sup>26</sup> or other difficulties in searching prior art is at odds with the Constitution's goal of furthering scientific achievement<sup>27</sup>, and is further contrary to Congressional intent when it ratified 35 U.S.C. § 101.

---

<sup>23</sup>W. Nordhaus, Invention, Growth and Welfare: A Theoretical Treatment of Technological Change 8 (1969); Barzel, Optimal Timing of Innovations, 50 Rev. Econ. Stat. 348, 354 (1968).

<sup>24</sup>Goodman, The Policy Implications of Granting Patent Protection to Computer Software: An Economic Analysis, 37 Vand. L. Rev. 147 (1984).

<sup>25</sup>In re Ruschig, 145 USPQ 275, 286 (CCPA 1965).

<sup>26</sup>See Parker v. Flook, 437 U.S. at 587-588, where the acting Commissioner of Patents and Trademarks argues that patent protection of software algorithms "will have a debilitating effect on the rapidly expanding computer 'software' industry, and will require him to process thousands of additional patent applications".

<sup>27</sup>Goodman, supra.

The Patent Office, in discharging its duties to the public, has commendably required applicants for patents to provide an adequate quid pro quo in exchange for the monopoly sought. It should be equally alert in protecting the rights of applicants who have legally and properly established such a right. To do otherwise would be to unjustly enrich the public at the expense of the inventor, a result we feel confident Congress could not have intended. In re Herr, 153 USPQ 548 (CCPA 1967).

CONCLUSION FOR 35 U.S.C. 101 REJECTION

In conclusion, there is no judicially determined exception to computer programs being non-statutory, and 35 U.S.C. § 101 specifically allows for the claimed subject matter, which is both a subcombination of a machine and an article of manufacture.

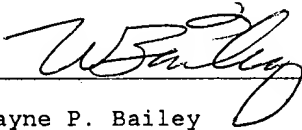
For the reasons advanced in this argument and summarized above, it is respectfully submitted that the Examiner erred in rejecting Claim 27 as being non-statutory under 35 U.S.C. § 101.

Applicants' attorney respectfully requests that the final rejection of Claims 1-27 be withdrawn and that the claims be passed to issue.

Respectfully Submitted,

R. A. Fabbio et al

By



Wayne P. Bailey

Attorney for Applicants

Registration No. 34,289

(512) 823-1012

WPB/bar



(7)

Appendix

1 Claim 1. An interface having means for presenting items for  
2 selection by a user of a data processing system, and having  
3 means for executing the selected items, said interface com-  
4 prising:

5 means for representing a plurality of interface objects  
6 in an object database; and

7 means for dynamically associating different ones of said  
8 interface objects into a plurality of logical frame presenta-  
9 tions based upon the data within each of said different ones  
10 of said interface objects.

1 Claim 2. The ~~interface of claim 1~~ wherein at least one of  
2 said plurality of interface objects represent at least one  
3 attribute of at least one system resource.

1 Claim 3. The interface of claim 1 wherein at least two of the  
2 plurality of interface objects represent a hierarchical  
3 relationship between components of the data processing system  
4 based upon the data within each of said at least two interface  
5 objects.

1 Claim 4. The interface of claim 2 wherein the interface  
2 objects are dynamically associated according to said hierar-  
3 chical relationship represented within each of said at least  
4 two interface objects.

1 Claim 5. The interface of claim 1 further comprising means  
2 for managing of a screen presentation of the objects and a  
3 user interaction with said objects based upon the data within  
4 at least one of the plurality of interface objects.

1 Claim 6. The interface of claim 2 further comprising means  
2 for utilizing a current value of said at least one attribute  
3 of said at least one system resource for presentation to said  
4 user.

1 Claim 7. The interface of claim 2 further comprising means  
2 for utilizing at least one instance of at least one of said  
3 system resources for presentation to said user for informing  
4 said user of an availability of said instance of said system  
5 resource.

1 Claim 8. The interface of claim 7 further comprising means  
2 for allowing the user to select said instance of said system  
3 resource presented to said user.

1 Claim 9. The interface of claim 2 further comprising means  
2 for utilizing a current value of said at least one attribute  
3 of said at least one system resource for a validation of a  
4 user response.

1 Claim 10. The interface of claim 1 further comprising means  
2 for dynamically constructing a command by associating at least  
3 one user input value with an option within said at least one  
4 of said interface objects.

1 Claim 11. The interface of claim 10 further comprising means  
2 for executing said constructed command.



1 Claim 12. The interface of claim 11 further comprising means  
2 for logging said executed command for later reexecution.

1 Claim 13. The interface of claim 1 further comprising means  
2 for constructing and executing a command based on a current  
3 state of said data processing system, a plurality of user  
4 selections, and the data within said interface objects.

1 Claim 14. The interface of claim 1 further comprising means  
2 for retrieving said objects from said object database in  
3 response to said user selected item.

1 Claim 15. The interface of claim 1 further comprising means  
2 for iteratively presenting said interface objects to said user  
3 dependent upon a plurality of user selections and said data  
4 within said interface objects.

1 Claim 16. The interface of claim 1 further comprising means  
2 for accessing at least one interface object from a plurality  
3 of screen presentations.

1 Claim 17. The interface of claim 1 further comprising means  
2 for accessing at least one screen presentation from a plural-  
3 ity of interface objects.

1 Claim 18. The interface of claim 1 further comprising means  
2 for altering the interface object database from within the  
3 interface during a session of execution of said interface, and  
4 means for reflecting said altered interface during said same  
5 session of execution of said interface.

1 Claim 19. The interface of claim 1 further comprising means  
2 for altering said interface object database by creating at  
3 least one new interface object.

1 Claim 20. The interface of claim 3 further comprising means  
2 for directly entering said hierarchy of interface objects at  
3 at least one of a plurality of locations within said hierar-  
4 chy.

1 Claim 21. The interface of claim 1 further comprising means  
2 for displaying said logical frame presentations by a plurality  
3 of graphical libraries.

1 Claim 22. The interface of claim 1 further comprising means,  
2 within said interface data objects, for representing said  
3 items in said logical frame presentation in at least one of a  
4 plurality of ways dependent upon a graphical context.

1 Claim 23. The interface of claim 1 further comprising means,  
2 within said interface data objects, for representing said  
3 items in said logical frame presentation in at least one of a  
4 plurality of ways dependent upon a linguistic context.

1 Claim 24. The interface of claims 1 further comprising means  
2 for accessing a screen library having means for indicating to  
3 said user a number of said items in said logical frame presen-  
4 tation currently outside of a visual screen presentation to  
5 said user.

1 Claim 25. The interface of claim 1 further comprising means  
2 for providing at least one logical frame presentation

3 dependent upon at least one access control policy applied to  
4 said plurality of interface objects.

1 Claim 26. A method for presenting items for selection by a  
2 user of a data processing system, said method comprising:

3 representing a plurality of interface objects in an  
4 object database; and

5 dynamically associating different ones of said interface  
6 objects into a plurality of logical frame presentations based  
7 upon the data within each of said different ones of said  
8 interface objects.

1 Claim 27. A computer program, residing on a computer compati-  
2 ble medium, having means for presenting items for selection by  
3 a user of a data processing system, said computer program  
4 comprising:

5 means for representing a plurality of interface objects  
6 in an object database; and

7 means for dynamically associating different ones of said  
8 interface objects with a plurality of logical frame  
9 presentations based upon data within each of said differ-  
10 ent ones of said interface objects.